(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 2 124 189 A1

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
25.11.2009 Bulletin 2009/48

(51) Int Cl.:
G06T 5/00 *(2006.01)*

(21) Application number: **08156447.8**

(22) Date of filing: **19.05.2008**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MT NL NO PL PT RO SE SI SK TR
Designated Extension States:
AL BA MK RS

(71) Applicants:
- **Mitsubishi Electric Information Technology Centre**
  Europe B.V.
  The Surrey Research Park
  Guildford
  Surrey GU2 7YD (GB)
  Designated Contracting States:
  GB
- **Mitsubishi Denki Kabushiki Kaisha**
  Chiyoda-ku
  Tokyo, 100-8310 (JP)
  Designated Contracting States:
  AT BE BG CH CY CZ DE DK EE ES FI FR GR HR HU IE IS IT LI LT LU LV MC MT NL NO PL PT RO SE SI SK TR

(72) Inventors:
- **Sibiryakov, Alexander**
  Guildford, Surrey GU2 7YD (GB)
- **Bober, Miroslaw**
  Guildford, Surrey GU2 7YD (GB)

(74) Representative: **HOFFMANN EITLE**
Patent- und Rechtsanwälte
Arabellastrasse 4
81925 München (DE)

(54) Image processing to enhance image sharpness

(57) Blurred image data is sharpened by converting three channels of RGB data into a single channel of intensity data, processing the intensity data to generate integral image data, applying a variable size filter to the integral image data to generate box-filtered data, calculating a gain factor for each pixel position in dependence upon the box-filtered data, the intensity data and the size of the filter used for that pixel position, and multiplying the original RGB data of each pixel by the gain factor for that pixel to generate sharpened RGB data. The size of the filter is selected at each pixel position in dependence upon an estimate of the local amount of blur. In this way, as the amount of blur changes, the filter size changes appropriately. By processing the integral image data to generate box-filtered data, a constant number of processing operations are required for image sharpening irrespective of the size of filter that is used.

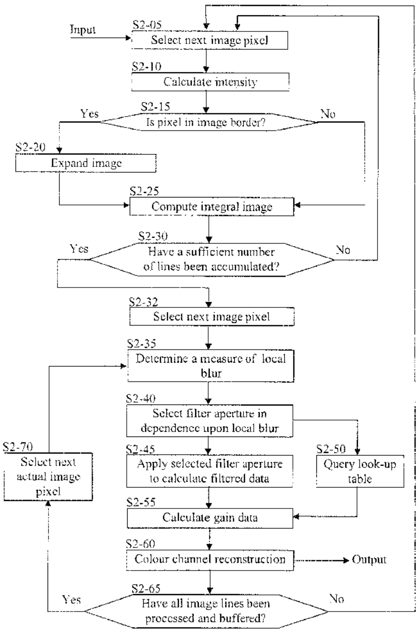
Input
S2-05
Select next image pixel
S2-10
Calculate intensity
S2-15
Yes
Is pixel in image border?
No
S2-20
Expand image
S2-25
Compute integral image
S2-30
Yes
Have a sufficient number of lines been accumulated?
No
S2-32
Select next image pixel
S2-35
Determine a measure of local blur
S2-40
Select filter aperture in dependence upon local blur
S2-70
Select next actual image pixel
S2-45
Apply selected filter aperture to calculate filtered data
S2-50
Query look-up table
S2-55
Calculate gain data
S2-60
Colour channel reconstruction
Output
S2-65
Yes
Have all image lines been processed and buffered?
No

Fig.2

Printed by Jouve, 75001 PARIS (FR)

Description

[0001] The present invention relates to the processing of images with blur to enhance the sharpness thereof.

[0002] Sharpening is one of the standard image processing techniques that are usually applied to visually enhance images. The effect of the sharpening usually appear very impressive to the users as it seems to bring out image details that were not there before. What sharpening actually does is to emphasize edges in the image and make them easier for the eye to pick out. No new details are created in the image.

[0003] The first step in sharpening an image is to blur it using one of many available prior art methods, e.g. pixel averaging, convolution with Gaussian mask or any other low-pass filtering. Next, the original image and the blurred version are processed so that if a pixel is brighter than the blurred version it is lightened further; if a pixel is darker than the blurred version, it is darkened. The result is to increase the contrast between each pixel and its neighbours. This process is usually called *Unsharp Masking* (UM) and can be represented by the following expression:

$$f_S(x,y) = f(x,y) + b(\ f(x,y) - f_{LP}(x,y)) \qquad (1)$$

[0004] In (1), $f(x,y)$ is the original image, $f_{LP}(x,y)$ is its low-pass filtered version and $f_s(x,y)$ is the result of sharpening. The boosting factor b determines how much the image difference is amplified.

[0005] The expression (1) can also be generalized by replacing the difference $f(x,y)$-$f_{LP}(x,y)$ by a general high-pass filter $f_{HP}(x,y)$:

$$f_S(x,y) = f(x,y) + b\ f_{HP}(x,y) \qquad (2)$$

[0006] The majority of prior art image sharpening methods use either form (1) or (2). The methods differ by the ways of choosing the boosting factor b and applying low-pass or high-pass filtering.

[0007] US patent 5,363,209 describes a variant of the UM method consisting of the following steps: 1) Converting the image to a luminance-chrominance format, wherein at least one signal represents overall image intensity; 2) Determining the maximum local contrast within the image; the local contrast is determined in 3x3-pixel neighbourhood; 3) Determining a 3x3 image filter, which increases maximum local contrast to a predetermined target value and all other contrast to an amount proportional thereto, and 4) Applying the determined filter function to the image to increase sharpness. Thus, in this method, $f_{LP}(x,y)$ is a 3x3 image average and b is a function depending on pixel position and image gradients: b=b $(x,y,f_x(x,y),f_y(x,y))$.

[0008] In the method described by Chiandussi and Ramponi in the paper entitled "Nonlinear Unsharp masking for the enhancement of Document Images", in the Proceedings of the Eighth European Signal Processing Conference, EUSIPCO-96, the structure of the UM method is maintained but a Laplacian-of-Gaussian (LoG) band-pass filter substitutes for the high-pass filter. The signal, which feeds the LoG filter, is processed by a quadratic operator which introduces a significant noise smoothing in uniform background areas. The overall structure of the algorithm corresponds to that of a homogeneous quadratic filter. The paper shows that quadratic filters are capable of edge enhancement in images with limited noise amplification.

[0009] Many image sharpening and deblurring methods have been developed in last two decades. The most powerful methods perform blind deconvolution of images when the shape and size of the convolution kernel are unknown. Such algorithms are usually used in off-line processing, when the image that requires enhancement has already been captured and stored in a memory or a hard disk. In off-line mode, the requirements of processing speed and amount of memory are not critical.

[0010] A number of problems exist with known image sharpening and deblurring methods.

[0011] For example, a first problem is that modern image sensors require real-time performance of the image enhancement algorithms. Miniaturization and low price of the sensors significantly constrain image processors, while keeping high requirements on quality of the result.

[0012] A second problem is that capturing of a non-flat subject results in variable blur in the image of that subject; the blur amount depends on distance from the current position of the sensor to the subject. The amount of blur affects the size (that is, the aperture) of the sharpening filter that is required to sharpen the image.

[0013] Prior art methods handle this problem by performing the sharpening algorithm a plurality of times, each time using a different size filter aperture and then fusing the resulting images. However, this is not efficient from the points of view of memory consumption and speed of computation.

[0014] Furthermore, prior art techniques do not concentrate on optimization, and are not suitable for direct embedded implementation and/or very low-cost implementation.

[0015] According to the present invention, there is provided, an apparatus and method for processing input image data to sharpen the image data. The image data is processed using a filter with a size that changes for different parts of the image in accordance with the amount of blur in that part. In this way, the filter size can be matched to the amount of blur in each different part.

[0016] In accordance with these features, the image data need only be filtered once, thereby reducing processing time and reducing processing resource requirements. In addition, by sharpening the image data with a filter having a size which is adaptively changed in accordance with changes in the blur of the image data, the sharpening is tailored for each part of the image.

[0017] The present invention also provides a computer program product, such as a storage medium or a signal, carrying computer program instructions to program a programmable processing apparatus to become operable to perform a method as set out above or to become configured as an apparatus as set out above.

[0018] Embodiments of the present invention provide a number of further advantages. In particular:

- An embodiment reduces the complexity of elementary operations. More particularly, an embodiment uses only integer operations in fixed-point format and all divisions are replaced by bit shifts.
- In an embodiment, a single buffer for storing a few image lines is sufficient for the proposed algorithm. Also, in-place processing is possible at no additional computational cost, when the output image is stored in the memory of the processed input image.

[0019] Embodiments of the invention can perform fast image sharpening, and are suitable for real-time implementation in a low-cost CPU embedded into linear image sensors.

[0020] Embodiments of the invention will now be described, by way of example only, with reference to the accompanying figures, in which:

Figure 1 schematically shows the components of an embodiment of the invention, together with the notional functional processing units into which the processing apparatus component may be thought of as being configured when programmed by computer program instructions;

Figure 2 shows the operations performed by the processing apparatus shown in Figure 1 to sharpen input image data; and

Figure 3 shows an illustrative representation of how data is stored within the apparatus of Figure 1 at various stages of the processing.

[0021] An embodiment will be described below in which processing is performed in an image scanner to sharpen the image data produced by a line sensor in the image scanner. It will be appreciated, however, that the processing may be performed in other types of apparatus, and that the processing may be performed on image data from other types of sensor, such as a full-image sensor.

[0022] As will be described below, the present embodiment uses a variable aperture of the sharpening filter depending on a locally estimated blur amount. The variable aperture is a function $r(x,y) \in [1..r_{max}]$ available at any pixel (x,y).

[0023] As a result, the filter aperture is selected at each pixel position in dependence upon the amount of blur in the image local to that pixel position. In this way, as the amount of blur in the image changes from one pixel position to another, the filter aperture size changes appropriately. This avoids the disadvantage of the prior art mentioned above which processes the whole image a plurality of times, each time using a filter with a different size aperture, and then fuses the results.

[0024] In addition, the embodiment produces the result progressively, as soon as new image lines are captured by the sensor. Thus, the embodiment requires a relatively small image processing buffer even in the case of the variable aperture of the filter. This is another difference from the prior art methods that usually work on an entire image requiring large intermediate memory for image processing.

[0025] Furthermore, the embodiment applies a filter to integral image data to generate box-filtered data, instead of applying a sharpening filter directly to intensity data. As a result, the number of processing operations required to perform the filtering operation remains constant, irrespective of the size of the filter aperture that is used to generate the box-filtered data.

[0026] Before describing the apparatus of the embodiment and the actual processing operations performed thereby, the processing algorithm employed in the embodiment will be described first.

[0027] In the image scanner of the present embodiment, the input image is a colour image consisting of three channels: R,G and B. In order to reduce computational load required for three channels, the embodiment process an intensity channel derived from the colour channels using a colour space transformation. Then, after image sharpening has been performed on the intensity channel, the three colour channels are reconstructed using the three original colour channels, the derived intensity channel before sharpening and the processed intensity channel after sharpening.

[0028] There are many conventional colour-to-intensity conversions that could be used to derive the intensity channel to be processed from the RGB channels. The present embodiment uses the simplified expression given by equation (3) because this requires only two additions:

$$f(x,y) = R(x,y) + G(x,y) + B(x,y) \qquad (3)$$

[0029] The low-pass filter included in the basic method (1) is replaced by a box filter f_{BF} , which is the sum of the pixels in a rectangular region *2r+1* x *2r+1*. The box filter is computed independently of region size r using a well-known integral image representation. The integral image I(*x,y*) is computed by recursive 2x2 filtering (that is, using four pixel references) as follows:

$$I(x,y) = f(x,y) + I(x-1,y) + I(x,y-1) - I(x-1,y-1) \qquad (4)$$

[0030] In the present embodiment, the variable aperture size *r*=*r*(*x,y*) is determined in dependence upon the amount of blur in the neighbourhood of the pixel being processed, with the amount of blur being provided by the image scanner. The box filter f_{BF} with variable aperture is computed using the integral image and only four memory references for any *r*, as follows:

$$f_{BF}(x,y,r) = I(x+r,y+r) - I(x-r-1,y+r) - I(x+r,y-r-1) + I(x-r-1,y-r-1) \qquad (5)$$

where *r*(*x,y*) is replaced by *r* for clarity.

[0031] The result of box filter (5) is used to obtain a "gain factor" *g*(*x,y*), which determine a multiplicative change of colour components:

$$g(x,y) = k + L_r[f(x,y)]\, f_{BF}(x,y,r) \qquad (6)$$

where *k* is a pre-computed numerical constant, depending on boosting factor in (1), $L_r[...]$ is a look-up-table pre-computed for each possible value $r(x,y) \in [1..r_{max}]$ and each value of intensity. The amount of memory required by this look-up-table is usually small, because the range of intensity values in equation (3) is [0..765], assuming byte range of the input colour channels.

[0032] The gain factor given by equation (6) is used to modify the input colours in other to obtain the output colours in accordance with equation (7) below, which are the final result of the algorithm:

$$C_s(x,y) = g(x,y)C(x,y), \quad \text{where} \quad C = R, G, \text{ or } B \qquad (7)$$

[0033] One of the simplest methods of colour reconstruction based on processed intensity is the multiplicative method, in which the colour component value is assumed to be proportional to intensity. The present embodiment uses the multiplicative method of colour reconstruction, given by equations (8) and (9):

$$g(x,y) = \frac{f_s(x,y)}{f(x,y)} \qquad (8)$$

$$C_s(x,y) = g(x,y)C(x,y), \quad \text{where} \quad C = R, G, \text{ or } B \qquad (9)$$

where *R*(*x,y*), *G*(*x,y*), *B*(*x,y*) and *f*(*x,y*) are original colours and intensity and $R_s(x,y)$, $G_s(x,y)$, $B_s(x,y)$ and $f_s(x,y)$ are modified colours and intensity.

[0034] The present embodiment performs image sharpening, as indicated by the subscript "s". The UM sharpening procedure (1) discussed previously can be represented in an equivalent form given by equation (10) indicating that the result of sharpening is a linear combination of the original image and its low-pass filtered version.

$$f_S(x,y) = (1+b)\ f(x,y) - b\ f_{LP}(x,y) \qquad (10)$$

[0035] From equations (8) and (10) a new representation (11) of the gain factor *g*(*x,y*) is:

$$g(x,y) = 1 + b - b\frac{f_{LP}(x,y)}{f(x,y)} \qquad (11)$$

[0036] As noted above, the present embodiment implements a general low-pass filter f_{LP} as the box filter f_{BF}, which is the simplest variant of low-pass filtering. The box filter of variable size *r*(*x,y*) is the pixel sum in a (2*r*+1) x (2*r*+1) rectangular region, given by

$$f_{BF}(x,y,r) = \sum_{i=-r}^{r}\sum_{j=-r}^{r} f(x+i, y+j) \qquad (12)$$

where *r*(*x,y*) is replaced by *r* for clarity. At each pixel, equation (12) requires $(2r+1)^2$ pixel references, which grows quadratically with an increase in the size of *r*.

[0037] In the present embodiment, the box filter uses a constant number of operations independently of the region size. This method is based on the representation of the image in the integral form *I*(x,y):

$$I(x,y) = \sum_{a\le x}\sum_{b\le y} f(a,b) \qquad (13)$$

from which the sum in equation (12) can be replaced by four references to the integral image:

$$f_{BF}(x,y,r) = I(x+r,y+r) - I(x-r-1,y+r) - I(x+r,y-r-1) + I(x-r-1,y-r-1) \qquad (14)$$

where *r*(*x*,*y*) is replaced by *r* for clarity. Thus, the present embodiment reduces the complexity of the sharpening with a variable aperture to the complexity of a simple 2x2 filter.

[0038] The definition given by equation (13) can also be replaced by an equivalent recursive definition given by equation (15) below, that also requires only four references to the image buffers:

$$I(x,y) = f(x,y) + I(x-1,y) + I(x,y-1) - I(x-1,y-1) \qquad (15)$$

[0039] Using the box filter, the computation of the gain factor (11) is performed according to equation (16) below, where low-pass filter is replaced by a local image average:

$$g(x,y) = 1 + b - \frac{b}{(2r+1)^2}\frac{f_{BF}(x,y,r)}{f(x,y)} \qquad (16)$$

[0040] The floating-point gain factor given by equation (16) is not convenient for low-cost embedded implementation. Therefore, the present embodiment uses a fixed-point format with a precision of *p* binary digits:

$$g_1(x,y) = 2^p \ g(x,y) \qquad (17)$$

[0041] For 32-bit integer representation, a large precision, e.g. *p*=22, can be used. In fixed-point representation the division operations in equation (16) can be performed in integer format. The present embodiment further reduces the complexity of the algorithm by excluding divisions from computations. More particularly, the present embodiment uses pre-computed look-up-tables $L_r[f(x,y)]$ for all possible values of the variable aperture $r \in [1..r_{max}]$ and intensity $f \in [0..765]$, assuming byte range of the input colour channels:

$$L_r[f] = -\frac{2^p b}{(2r+1)^2 f(x,y)} \qquad (18)$$

[0042] The final expression of the gain factor in fixed-point format is given by equation (19)

$$g(x,y) = k + L_r[f(x,y)] \ f_{BF}(x,y,r) \qquad (19)$$

where $k=2^p(1+b)$.

[0043] After applying the colour reconstruction defined by equation (9), the obtained colour components are also represented in the fixed-point format. To return to the original range, equation (9) is divided by 2^p, which is efficiently implemented in the present embodiment by bit shifting to the right by *p* bits ('>>' operation in C/C++):

$$C_s(x,y) = g(x,y)C(x,y) \ \texttt{>>} \ p, \quad \text{where} \quad C = R, G, \text{ or } B \qquad (20)$$

[0044] Finally, the output values $C_s(x,y)$ are clipped in order to fit into the original range, e.g. [0..255].

[0045] The present embodiment reduces the amount of intermediate memory required to perform image sharpening.

[0046] More particularly, in the computational procedure given by equations (3)-(7) above, only the integral image given by equation (4) needs additional memory buffer; all other values, such as intensity *f*(*x*,*y*), box filter $f_{BF}(x,y,r)$ and gain factor $g(x,y)$ are used immediately after computing them. From the analysis of equations (6) and (7) one can conclude that the gain factor $g(x,y)$ does not need storage for all pixels (x,y); $g(x,y)$ can be stored in a local variable g, which is used to modify the colours given by equation (7) at the current pixel. Similarly, $f_{BF}(x,y,r)$ does not require any variable, as it can be directly substituted to equation (6). The intensity does not require a memory buffer due to simplicity of the equation (3); it can be recomputed in both the integral image given by equation (4) and the gain factor given by equation (6).

[0047] It will be seen from equation (14) that the box filter is not defined near the edges of the image, because pixel coordinates $(x \pm r, y \pm r)$ can be outside the image. The present embodiment therefore expands the image so that the computation of box filter defined by equation (14) can be performed correctly at the edges and corners of the integral image. The preferred embodiment uses simple pixel replication, which means that each pixel of the expanded image lying outside the original image has a value equal to that of the nearest pixel of the original, non-expanded image. This expansion is defined for the outside pixels as follows:

$$\begin{array}{lll}
1)\ f(-x,-y) = f(0,0), & \text{for } 0 < x \le r_{max}+1, & 0 < y \le r_{max}+1 \\
2)\ f(x,-y) = f(x,0), & \text{for } 0 \le x < M, & 0 < y \le r_{max}+1 \\
3)\ f(M+x,-y) = f(M-1,0), & \text{for } 0 < x \le r_{max}, & 0 < y \le r_{max}+1 \\
4)\ f(-x,y) = f(0,y), & \text{for } 0 < x \le r_{max}+1, & 0 \le y < N \\
5)\ f(M+x,y) = f(M-1,y), & \text{for } 0 < x \le r_{max}, & 0 \le y < N \\
6)\ f(-x,N+y) = f(0,N-1), & \text{for } 0 < x \le r_{max}, & 0 < y \le r_{max} \\
7)\ f(x,N+y) = f(x,N-1), & \text{for } 0 \le x < M, & 0 < y \le r_{max} \\
8)\ f(M+x,N+y) = f(M-1,N-1), & \text{for } 0 < x \le r_{max}, & 0 < y \le r_{max},
\end{array} \qquad (21)$$

where M and N are the width and height of the original image respectively.

[0048] Using the definitions in equation (21), the integral image defined by equation (15) is computed for all pixels from the extended range defined by equation (22) below.

$$\begin{array}{l}
x \in [-r_{max},\ M + r_{max}-1] \\
y \in [-r_{max},\ N + r_{max}-1]
\end{array} \qquad (22)$$

[0049] After this, the box filter can be computed correctly for all pixels locations.

[0050] In order to sharpen one image line, the algorithm uses an image neighbourhood consisting of $2r_{max}+2$ lines (including the current line, $r_{max}+1$ previous lines and r_{max} next lines). This neighbourhood is used in the box filter defined by equation (5). Thus, the internal memory buffers should have a capacity to store pixel values for at least N_r x d pixels, where N_r is the expanded image width N so that $N_r=N+2r_{max}+1$ and $d_{max}=2r_{max}+2$, is the neighbourhood size used in the sharpening filter. At the initial stage, the algorithm processes r_{max} lines of the original image and fills the internal buffer with the intermediate result of processing, which is the integral image I. When a new line with number $i=r_{max}+1$, $r_{max}+2,\ldots, M$ is received from the scanner, the algorithm updates the internal buffer with the new line of the integral image I, and performs the sharpening algorithm for the line with number $i-r_{max}$. At the final stage, when all M lines of the image are scanned, and $M-r_{max}$ lines are sharpened, the algorithm performs sharpening of the remaining r_{max} lines.

[0051] Having provided a summary of the algorithm employed in the present embodiment, the apparatus of the embodiment and the processing operations performed thereby will now be described.

[0052] Referring to Figure 1, an embodiment of the invention comprises a programmable processing apparatus 2, containing, in a conventional manner, one or more processors, memories, graphics cards etc.

[0053] The processing apparatus 2 is programmed to operate in accordance with programming instructions input, for example, as data stored on a data storage medium (such as an optical CD Rom 4, semiconductor ROM, magnetic recording medium 6, etc), and/or as a signal 8 (for example an electrical or optical signal input to the processing apparatus 2, for example from a remote database, by transmission over a communication network such as the Internet, or by transmission through the atmosphere).

[0054] When programmed by the programming instructions, processing apparatus 2 can be thought of as being configured as a number of functional units for performing processing operations. Examples of such functional units and their interconnections are shown in Figure 1. The units and interconnections illustrated in FIG.1 are, however, notional, and are shown for illustration purpose only to assist understanding; they do not necessarily represent units and connections into which the processor, memory etc of the processing apparatus 2 actually become configured.

[0055] Referring to the functional units shown in Figure 1, the RGB data storage section 10 is configured to receive and store input image data from the line sensor of the scanner (not shown), in the form of red, green and blue intensity values for each pixel. As noted above, this data is received in three channels, namely a red, green and blue channel. In the present embodiment, the RGB data storage section 10 is also configured to store sharpened image data generated by the process apparatus 2, prior to output. This avoids the need for an additional memory to store the sharpened image data.

[0056] The local blur estimate storage section 12 is configured to receive and store an estimate of the local blur for each pixel of the image from the image scanner.

[0057] The image scan controller 14 is configured to control the flow of data to and from the RGB data storage section 10 during processing. In use of the apparatus, the image scan controller 14 determines, inter alia, which pixel to process next.

[0058] The variable aperture controller 16 is configured to control the size of the aperture of the filter employed to process each pixel in the input image data, in dependence upon the estimate of local blur stored for the pixel in the local blur estimate storage section 12.

[0059] The pre-computed look-up tables 18 store pre-computed values of the variable $L_r[f]$ in equation (18) as function of image data intensity and filter aperture size. During the sharpening process, the pre-computed look-up tables are addressed using calculated values of image data intensity and filter aperture size to output a value of said variable without the need to evaluate equation (18), which includes computationally costly division calculations. Accordingly, the pre-computed look-up tables 18 remove the need to carry out division operations during the image sharpening process, thereby reducing processing requirements and speeding up processing.

[0060] The image data processing buffer 20 is configured to store the intermediate results produced during the image sharpening process, such as the integral data generated in accordance with equation (19) above.

[0061] The image sharpener 22 is configured to process image data, received from the RGB data storage section 10, to provide sharpened image data.

[0062] In the present embodiment, the image sharpener 22 comprises:

- an intensity calculator 24 for calculating a single intensity channel from the input image data stored in the RGB data storage section 10;
- an image expander 26 for generating an expanded image based on the input image prior to processing so as to reduce unwanted edge effects in the resulting sharpened image data;
- an integral image calculator 28 for generating integral image data from the expanded image generated by the image expander 26, and providing the same to the image data processing buffer 20;
- a variable aperture filter 30 for processing integral image data provided by image data processing buffer 20, to generate filtered image data;
- a gain factor calculator 32 for determining a gain factor based on the product of the filtered image data provided by the variable aperture filter 30 and the output of the pre-computed look-up tables 18; and
- an output colour channel calculator 34 for modulating the input image data stored in the RGB data storage section 10 by the gain factor determined by the gain factor calculator 32.

[0063] The output image data interface 36 is configured to communicate sharpened RGB pixel data, stored in the RGB data storage section 10, to a further component of the scanner or an external device.

[0064] Figure 2 shows the processing operations performed by processing apparatus 2 to process input data in this embodiment.

[0065] Referring to Figure 2, in step S2-05, the image scan controller 14 selects the first (if the process has just been initiated) or the next image pixel. The RGB data storage section 10 then provides data for the selected pixel in each of the three colour channels to the image sharpener 22.

[0066] In step S2-10, intensity calculator 24 processes the RGB data from the RGB data storage section 10 to a generate a single intensity channel.

[0067] There exist many luminance-chrominance colour spaces, such as HSV, YCrCb, L*u*v*, L*a*b*, providing an intensity or luminance channel. All these colour spaces differ by the number of operations required to compute the intensity f from the colour components R,G and B. The present embodiment generates the intensity channel using the simplest intensity representation given by equation (3) above in integer format, thereby requiring by only two additions. However, any other more complex transformation of the colour to intensity $f(x,y) = f(R(x,y),G(x,y),B(x,y))$ can be used instead.

[0068] The result of the processing at step S2-10 is a single channel of data on which image sharpening is to be performed. This avoids the need to perform image sharpening of each of the R, G and B channels, thereby reducing processing requirements and/or time.

[0069] As noted previously, in order to avoid unwanted effects at the borders of the resulting sharpened image, it is necessary to extend the image prior to processing. Accordingly, in step S2-15, the image scan controller 14 determines whether the current pixel is located in a border of the input image which, in the present embodiment, has a depth of one pixel.

[0070] If the current pixel does lie in a border, then the image sharpener 22 carries out image expansion in step S2-20 using the image expander 26. This image expansion is performed in accordance with equation (21) above, producing intensity data for the expanded regions of the image, which is stored in the image data processing buffer 20.

[0071] Following image expansion, or if the current pixel is not a border pixel, integral image calculator 28 processes the intensity data to calculate integral image date in step S2-25. This processing is performed in accordance with equation (15) above.

[0072] In step S2-30, image scan controller 14 checks the number of lines of data that are stored in the image data processing buffer 20. More particularly, because the variable aperture filtering requires data from a neighbourhood centred on the currently selected pixel, it is only after a sufficient number of lines have been accumulated in the image data processing buffer 20, that the variable aperture filtering can commence. Accordingly, in step S2-30, the image scan controller 14 determines if a sufficient number of lines (comprising $2r_{max} + 2$ lines) have been accumulated in the image data processing buffer 20.

[0073] If a sufficient number of lines have not been accumulated, processing returns to step S2-05 and the processing at steps S2-05 to S2-30 described above is repeated until a sufficient number of lines have been accumulated in the image data processing buffer 20.

[0074] When a sufficient number of lines have been accumulated in the image data processing buffer 20, then the processing proceeds to step S2-32.

[0075] At step S2-32, the image scan controller 14 selects the next input image pixel for which sharpened image data has not yet been calculated.

[0076] In step S2-35, the variable aperture controller 16 determines a measure of the local image blur corresponding to the selected image pixel. In the present embodiment this is done by reading the estimated blur provided by the scanner and stored in the local blur estimate storage section 12 for the current pixel. One way in which the scanner may provide the local blur estimate is described in our co-pending patent application entitled "Document Scanner" (attorney reference 128 250) filed concurrently herewith, the entire contents of which are incorporated herein by cross-reference. However, if the processing apparatus 2 is employed within an apparatus which does not provide an estimate of blur, or if the processing apparatus 2 is a stand-alone apparatus, then the variable aperture controller 16 may itself perform processing to calculate a blur measure using a conventional technique, such as one of those described in US 5,363,209.

[0077] In step S2-40, the variable aperture controller 16 selects an aperture for the filter to be applied to the integral image data in dependence upon the estimate of local blur determined at step S2-35. It will be appreciated that the method of selecting the size of the filter aperture is dependent upon the method used to measure the local blur and hence the values that the local blur will take. In the present embodiment, data is stored defining a respective aperture size for each range of the blur values produced for typical input images by the scanner. This stored data is generated in advance by testing images to determine the typical blur values thereof and assigning filter apertures to different ranges of these values. The maximum aperture size r_{max} is also assigned in this way.

[0078] In step S2-45, a filter with an aperture of the size selected in step S2-40 is applied to the integral image data to generate box-filtered image data in accordance with equation (14) above. The effect of this processing is the same as applying the box filter directly to the intensity data in accordance with equation (12) above. However, by applying the filter to the integral image data, a filtered value of the intensity data is obtained using a constant number of processing operations, irrespective of the size of the filter aperture. More particularly, the filtered image data is obtained with a number of processing operations equivalent to those required for a filter of size 2x2, as described previously.

[0079] In step S2-50, the size of the filter aperture selected in step S2-40 is used as one of two input values to query the pre-computed look-up tables 18. The other input value is the intensity of the image data for the current pixel. This was computed previously at step S2-10. However, in order to reduce memory requirements, the value computed at step S2-10 is not stored in the present embodiment, and instead it is recalculated at step S2-50 by processing the RGB data stored in the RGB data storage section 10 in the same way as at step S2-10.

[0080] As noted above, the use of the pre-computed look-up tables 18 is this way enables a value of $L_r[f]$ in equation (18) above to be computed without computationally expensive division calculations.

[0081] In step S2-55, the gain factor calculator 32 calculates a value for the gain "g" in accordance with equation (6) above using the value $L_r[f]$ read from the look-up tables 18 at step S2-50 and the filtered data produced at step S2-45.

[0082] In step S2-60, the output colour channel calculator 34 modulates the RGB data for the current pixel stored in the RGB data storage section 10 by the gain "g" calculated at step S2-55 to generate sharpened RGB data for the current pixel in accordance with equation (7) above. The sharpened RGB data values are then written back into the RGB data storage section 10, overwriting the original RGB values for the current pixel, pending output from the processing apparatus 2.

[0083] In step S2-65, the image scan controller 14 determines whether all of the lines of the input image have been converted to integral image data and buffered in the processing at steps S2-05 to S2-30.

[0084] If it is determined that not all of the lines of the input image have been processed in this way, then the processing returns to step S2-05, and the processing described above is repeated.

[0085] On the other hand, when it is determined that all of the lines of the input image have been converted to integral image data and buffered, then processing returns to step S2-35 via step S2-70 (at which the next pixel to be processed is selected). Steps S2-35 to S2-60 are then repeated in this way for each pixel that has not yet been processed to calculate a corresponding gain factor.

[0086] Figure 3 shows schematically the storage of data during the processing operations described above. More particularly, Figure 3 shows the storage of RGB data for the input image in RGB data storage unit 10, the storage of the integral image data (generated in step S2-25) in the image data processing buffer 20, and the storage of the sharpened image data (generated at step S2-60) in the RGB data storage section 10.

[0087] Referring to Figure 3, the effect of the processing at steps S2-05 to S2-30 to accumulate integral image data in the image data processing buffer 20 is represented as a "sliding buffer". More particularly, the position of the sliding buffer in Figure 3 schematically represents the different parts of the integral image for which data is stored in the image data processing buffer 20.

[0088] When the first line 30 of the input image is scanned, the sliding buffer has position 131. The first line is replicated in the buffer according to the first five rules in equation (21). During scanning of the next r_{max}-1 lines, the sliding buffer remains in the position 131 and pixels are replicated according to rules (4) and (5) in equation (21). As a result of this processing, replicated pixel values are generated in the shaded region shown in Figure 3 at position 131.

[0089] After the first r_{max} lines have been scanned, the integral image data in the sliding buffer is ready for the processing at steps S2-35 to S2-65 to apply the box filter and produce the first line 133 of the sharpened image data. Thus, a delay of r_{max} lines is created between the scanning of the RGB input mage data and the generation of the sharpened RGB data.

[0090] During subsequent scanning 134, the sliding buffer is moved to a new position 135. This is implemented by a cyclic shift of the pointers to the buffer's lines. In the new position 135, rules (4) and (5) from equation (21) are applied to replicate pixels in the shaded areas of position 135 shown in Figure 3.

[0091] At the final line 136 of the input image, the sliding buffer is in position 137, and rules (4) - (8) from equation (21) are applied to replicate the pixels.

[0092] During this scanning, there is always a delay of r_{max} lines between the input and output lines, which makes it possible to use the in-place processing described above, in which the sharpened image data for the output line is written directly to RGB data storage section 10 to replace the data of the input line.

Modifications and Variations

[0093] Many modifications and variations can be made to the embodiment described above.

[0094] For example in the embodiment above, the processing apparatus 2 forms part of an image scanner. However, instead, the processing apparatus 2 may be a stand-alone apparatus, such as a personal computer, or it may form part of a different type of apparatus, such as a digital camera, copier or printer.

[0095] Although the embodiment described above processes image data from a linear image sensor, the method of processing images disclosed hereinabove is equally applicable to image data captured using a rectangular array of image sensors, such as a CCD camera.

[0096] In the embodiment described above, a filter of variable size was applied to integral image data in order to generate box-filtered image data. However, neither the integral image data nor the box-filtering is essential. For example, instead, a variable size low pass filter, or a variable size high pass filter, could be applied to image data, such as intensity data, to generate sharpened image data.

[0097] The embodiment described above performs in-place processing to write the sharpened image data back into the RGB data storage section 10 overwriting the original RGB data of the input image. However, instead, a different memory may be provided to store the sharpened image data.

[0098] Although embodiments of the invention have been described above with reference to an input image represented as three colour channels with a range of intensity values of 0 to 255, it would be obvious to the skilled person that embodiments of the invention may be provided, which process an input image represented using any number of channels with any range of intensity values.

[0099] Although the embodiment above is configured to process colour input data, it may instead be configured to process black and white input data. In this case, the intensity calculator 22 would no longer be required because a single channel of intensity data of the black and white data would already be available for processing.

[0100] In the embodiment described above, processing is performed by a programmable processing apparatus using processing routines defined by computer program instructions. However, some, or all, of the processing could be performed using hardware instead.

[0101] Other modifications are, of course, possible.

Claims

1. A method of processing image data to generate sharpened image data, the method comprising:

   applying a filter to the image data to generate filtered image data, such that, for each of a plurality of pixel positions within the image data, a size of the filter to be applied is selected in dependence upon a measure of the image blur for that pixel position (S2-40), and a filter of the selected size is applied to generate filtered image data (S2-45); and
   modulating the image data in dependence upon the filtered image data (S2-60) to generate sharpened image data.

2. A method according to Claim 1, wherein:

   the process of modulating the image data to generate sharpened image data (S2-60) comprises calculating a gain factor in dependence upon the filtered image data and modulating the image data in dependence upon the calculated gain factor; and
   the gain factor is calculated for each of a plurality of pixel positions within the image data by:

      reading a stored value from a look-up table in dependence upon both the size of the filter applied to the image data for that pixel position and an intensity of the image data for that pixel position (S2-50); and
      calculating the gain factor as a function of the value read from the look-up table and the filtered image data for that pixel position (S2-55).

3. A method according to Claim 1 or Claim 2, wherein:

   the image data is accumulated in a memory for pixels on a number of lines of the image, wherein the number of lines is dependent upon the size of the filter to be applied (S2-30); and
   the processes of applying the filter (S2-45) and modulating the image data (S2-60) are carried out for pixels on a stored line before image data for pixels on a further line of the image is stored in the memory and the processes are repeated for pixels on the next stored line.

4. A method according to any preceding claim, wherein the image data to be sharpened is stored in a memory and overwritten in the memory with the sharpened image data as the sharpened image data is generated.

5. A method according to any preceding claim, wherein:

   the image data to be sharpened comprises colour data in a plurality of channels;
   the colour data is processed to generate intensity data (S2-10);
   the generated intensity data is processed to generate the filtered image data (S2-45); and
   the colour data in the plurality of channels is modulated in dependence upon the filtered image data (S2-60).

6. Apparatus for processing image data to generate sharpened image data, the apparatus comprising:

   filter selection means (16) for selecting, for each of a plurality of pixel positions within the image data, a respective size of the filter to be applied thereat in dependence upon a measure of the image blur for that pixel position;
   means (30) for filtering the image data by applying filters of the selected sizes to the image data to generate filtered image data; and
   modulation means (32, 34) for modulating the image data in dependence upon the filtered image data to generate

sharpened image data.

7. Apparatus according to Claim 6, wherein:

the modulation means (32, 34) comprises gain calculation means (32) for calculating a gain factor in dependence upon the filtered image data and means (34) for modulating the image data in dependence upon the calculated gain factor; and
the gain calculation means (32) is arranged to calculate the gain factor for each of a plurality of pixel positions within the image data by:

reading a stored value from a look-up table (18) in dependence upon both the size of the filter applied to the image data for that pixel position and an intensity of the image data for that pixel position; and
calculating the gain factor as a function of the value read from the look-up table and the filtered image data for that pixel position.

8. Apparatus according to Claim 6 or Claim 7, wherein:

the apparatus further comprises a memory and the apparatus is arranged to accumulate the image data for pixels on a number of lines of the image in the memory, wherein the number of lines is dependent upon the size of the filter to be applied; and
the means (30) for filtering the image data and the modulation means (32, 34) are arranged to process the image data for pixels on a stored line before image data for pixels on a further line of the image is stored in the memory and the processes are repeated for pixels on the next stored line.

9. Apparatus according to any of Claims 6 to 8, wherein the apparatus further comprises a memory for storing the image data to be sharpened and the apparatus is arranged to overwrite the data in the memory with the sharpened image data as the sharpened image data is generated.

10. Apparatus according to any of Claims 6 to 9, wherein:

the image data to be sharpened comprises colour data in a plurality of channels;
the apparatus further comprises means (24) for processing the colour data to generate intensity data;
the means (30) for filtering the image data is arranged to apply filters of the selected sizes to the generated intensity data to generate the filtered image data; and
the modulation means (34) is arranged to modulate colour data in the plurality of channels in dependence upon the filtered image data.

11. A computer program product carrying computer program instructions to program a programmable processing apparatus to become operable to perform a method as set out in at least one of Claims 1 to 5.

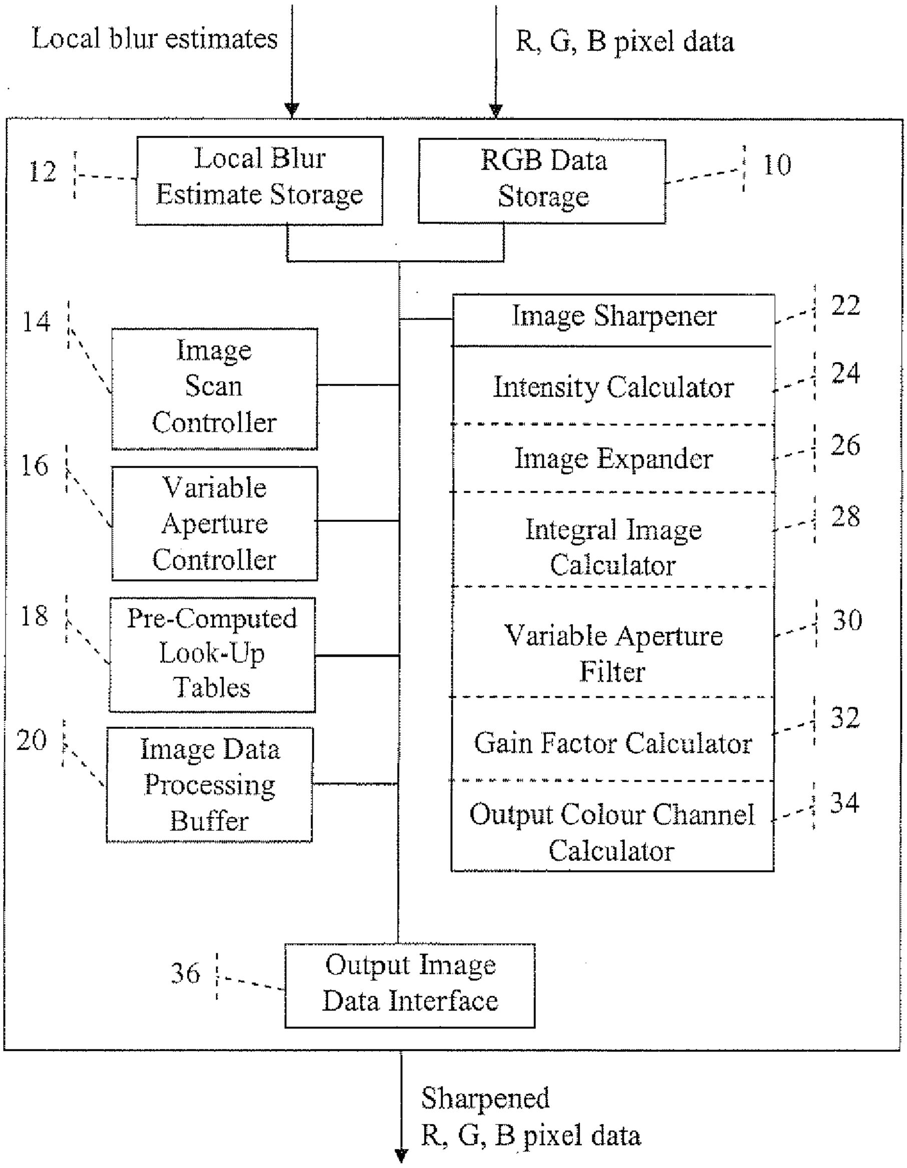
Local blur estimates
R, G, B pixel data
12
Local Blur Estimate Storage
RGB Data Storage
10
14
Image Scan Controller
16
Variable Aperture Controller
18
Pre-Computed Look-Up Tables
20
Image Data Processing Buffer
Image Sharpener
22
Intensity Calculator
24
Image Expander
26
Integral Image Calculator
28
Variable Aperture Filter
30
Gain Factor Calculator
32
Output Colour Channel Calculator
34
36
Output Image Data Interface
Sharpened R, G, B pixel data

Fig.1

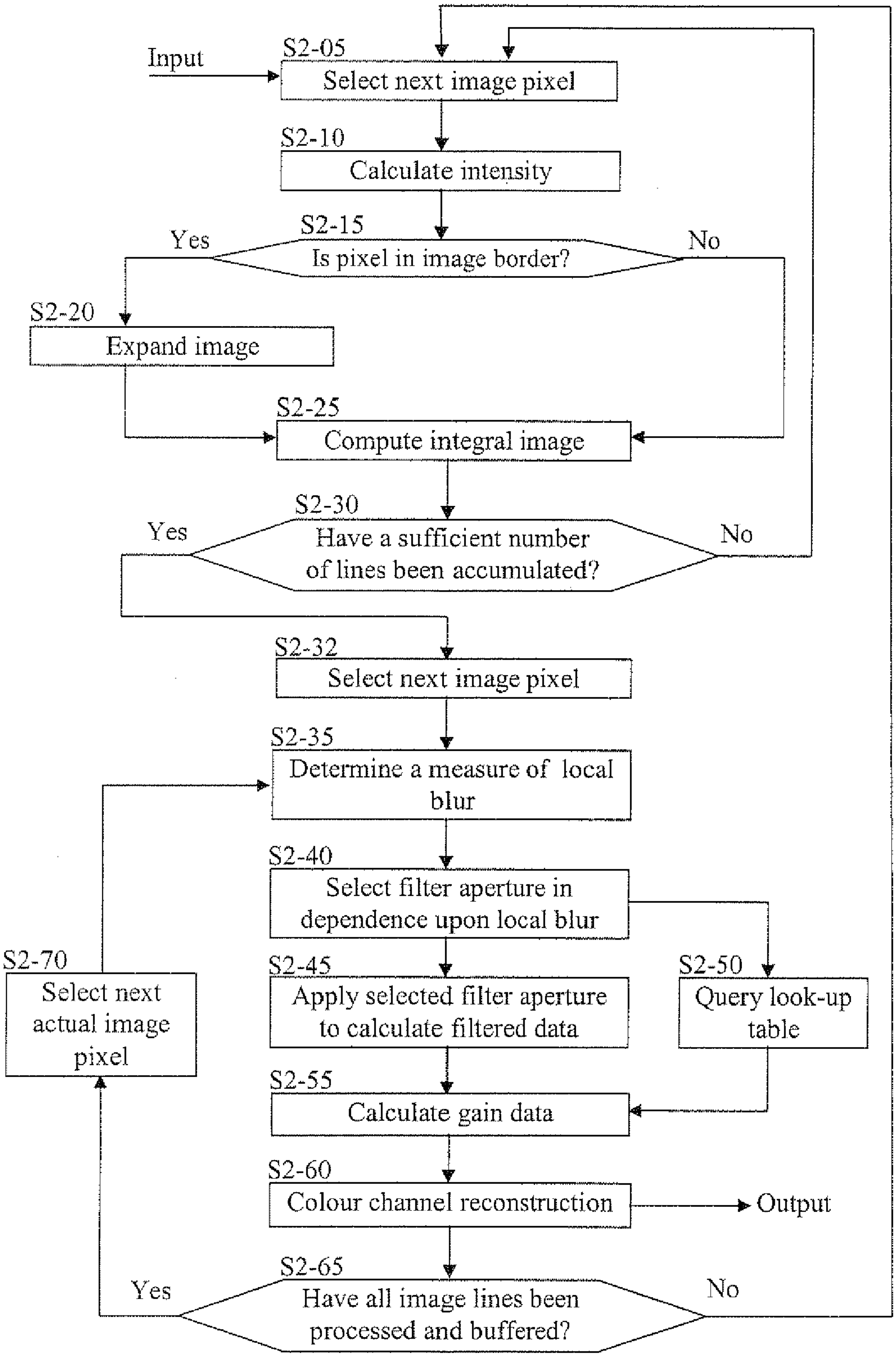
Input
S2-05
Select next image pixel
S2-10
Calculate intensity
S2-15
Is pixel in image border?
Yes
No
S2-20
Expand image
S2-25
Compute integral image
S2-30
Have a sufficient number of lines been accumulated?
Yes
No
S2-32
Select next image pixel
S2-35
Determine a measure of local blur
S2-40
Select filter aperture in dependence upon local blur
S2-45
Apply selected filter aperture to calculate filtered data
S2-50
Query look-up table
S2-55
Calculate gain data
S2-60
Colour channel reconstruction
Output
S2-65
Have all image lines been processed and buffered?
Yes
No
S2-70
Select next actual image pixel

Fig.2

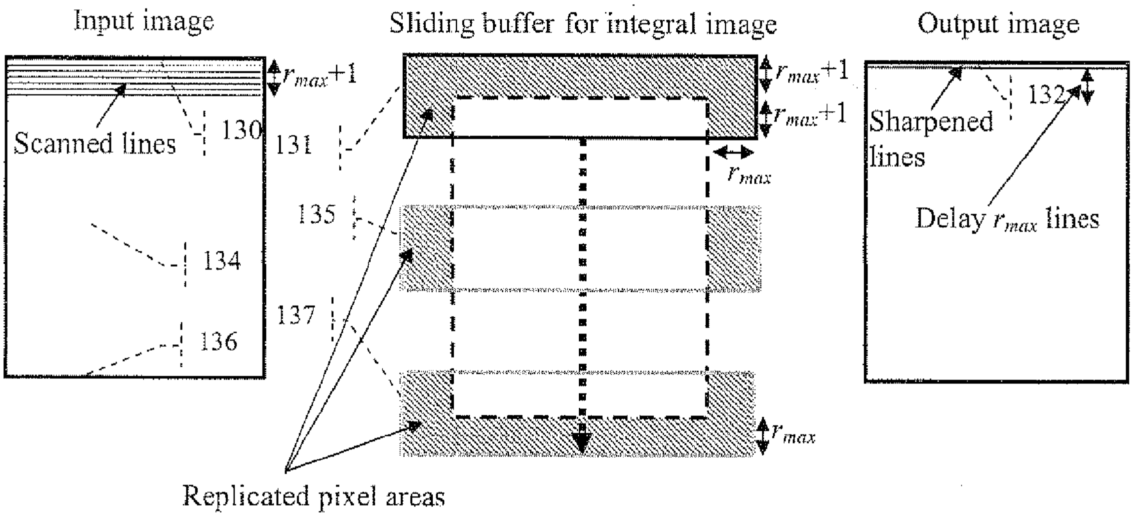
Input image
Sliding buffer for integral image
Output image
r_{max}+1
r_{max}+1
r_{max}+1
Scanned lines
130
131
r_{max}
Sharpened lines
132
Delay r_{max} lines
135
134
137
136
r_{max}
Replicated pixel areas

Fig. 3

European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 08 15 6447

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X Y	US 2005/249429 A1 (KITAMURA YOSHIRO [JP]) 10 November 2005 (2005-11-10) * paragraph [0181] - paragraph [0187] * -----	1,4-6, 9-11 2,3,7,8	INV. G06T5/00
Y	US 2002/006230 A1 (ENOMOTO JUN [JP]) 17 January 2002 (2002-01-17) * paragraph [0080] - paragraph [0123] * -----	2,7	
Y	PAUL WYATT ET AL: "Fast Feature Extraction Using Approximations to Derivatives with Summed-Area Images" COMPUTER VISION - ACCV 2006 LECTURE NOTES IN COMPUTER SCIENCE, LNCS, SPRINGER-VERLAG, BE, vol. 3851, 1 January 2005 (2005-01-01), pages 776-786, XP019027409 ISBN: 978-3-540-31219-2 * chapter 3 * -----	3,8	
A	US 2008/027994 A1 (GUAN HAIKE [JP]) 31 January 2008 (2008-01-31) * paragraph [0048] - paragraph [0085] * -----	1-11	TECHNICAL FIELDS SEARCHED (IPC)
			G06T

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Munich	1 August 2008	Rockinger, Oliver

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons
....
& : member of the same patent family, corresponding document

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 08 15 6447

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

01-08-2008

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2005249429 A1	10-11-2005	NONE	
US 2002006230 A1	17-01-2002	JP 4081219 B2 JP 2001298619 A	23-04-2008 26-10-2001
US 2008027994 A1	31-01-2008	EP 1885136 A2 JP 2008061217 A	06-02-2008 13-03-2008

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 5363209 A **[0007] [0076]**